



Nebenläufige Programmierung

Wintersemester 2004/05

Serie 7 (EST)

24. Januar 2005

Thema: Endsemestertest

Ausgabetermin: 24. Januar 2005

Abgabe: 7. Februar 2005 (12:00)

Diese Serie soll in Einzelarbeit gelöst werden! Auch Abschreiben lassen führt zur Disqualifikation. Insbesondere hat keine der 3 Personen, die im Mittsemestertest abgeschrieben haben, sich gemeldet (siehe Homepage).

Aufgabe 1 (4 Punkte) Exercise 2.29 aus Andrews auf S. 89.

Aufgabe 2 (4 Punkte) Gegeben seien $n > 0$ Prozesse, wobei jeder Prozess eine individuelle Information hat. Beweisen Sie formal, dass mindestens $2 \cdot (n - 1)$ one way message passing nötig sind, so dass jeder die Information des anderen weiß. D.h. zeigen Sie, dass es keine bessere Strategie als die zentralisierte- oder die Ring-Lösung (Andrews S. 317) gibt, um die Anzahl der Verbindungen zu reduzieren.

Tipp: Nehme kleinstes n an, wo es scheitert und zeige dass es dann auch schon für kleinere n scheitert.

Aufgabe 3 (4 Punkte) [*Message passing using semaphores*]

Sei `synch_send` ein Kommando für synchrones Senden und `receive` das übliche (synchrone) Empfangen. Beide Kommandos sind also blockierend, d.h. jedes der Kommandos terminiert erst, nachdem das Gegenstück aufgerufen und die Nachricht ausgetauscht wurde. Der Kanal soll als Mailbox-Kanal aufgefasst werden, d.h. beliebig viele Prozesse dürfen auf dem Kanal senden und empfangen.

Benutze Semaphoren (für wechselseitigen Ausschluss und zum Signalisieren) und entwickle eine Implementierung (Pseudocode) von `synch_send(msg)` und `receive(msg)`. Nimm an, dass es einen Pufferplatz der Größe 1 gibt, um die Nachricht vom Sender zum Empfänger zu kopieren. Minimieren Sie die Anzahl der Semaphoren (ohne Busy Waiting zu benutzen).

Aufgabe 4 (4 Punkte) Exercise 7.21 aus Andrews auf S. 358.

Aufgabe 5 (4 Punkte) Exercise 8.8 aus Andrews auf S. 417 (kein lauffähiges Programm nötig).